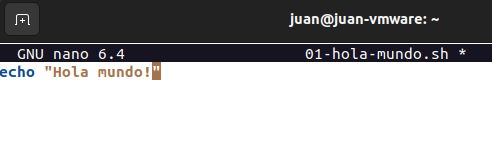


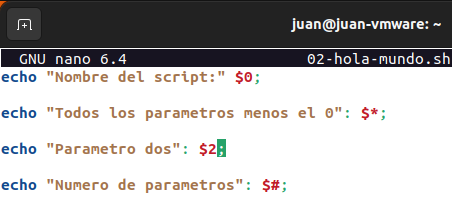
|  |
| --- |
|  |
| SHELL: SCRIPTS |
| IMPLANTACIÓN DE SISTEMAS  JUAN CARLOS NAVIDAD GARCÍA |

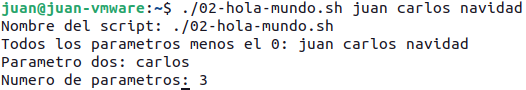
1. **Realizar un script llamado ‘01-hola-mundo.sh’ que muestre por pantalla: “Hola mundo!”.**

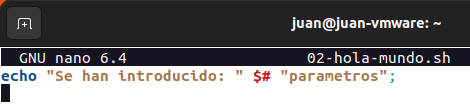
1. **Realizar un script llamado (‘02-hola-parametros.sh’) que muestre:**

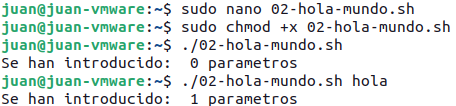
NOTA: Ejecuta el script con 3 parámetros **posicionales**

* 1. Nombre del script.
  2. Todos los parámetros introducidos.
  3. El valor del segundo parámetro introducido.
  4. El número de parámetros introducidos.

****

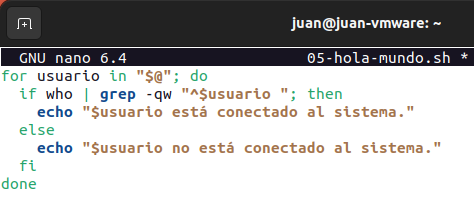
****

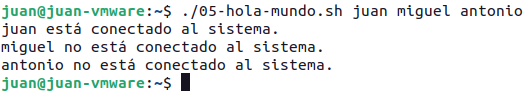
1. **Realizar un script llamado ‘03-hola-al-menos-1-parametro.sh” que verifique que al menos hayamos introducido un parámetro.**

****

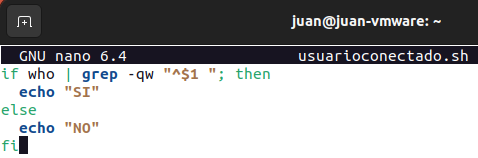
1. **Ídem y que además separe cada argumento por “, “ (‘04-hola-parametros-separados.sh’).**

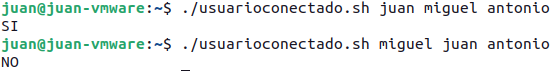
****

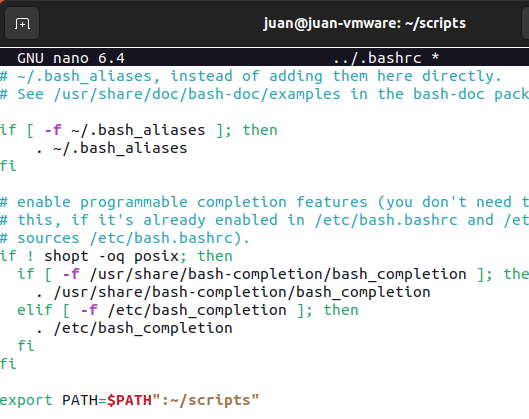
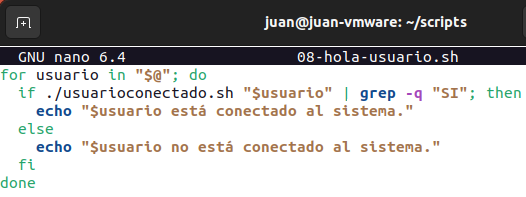
1. **Realiza un script que verifique que los parametros pasados sean usuarios conectados al sistema (‘05-hola-usuario.sh’).**

****

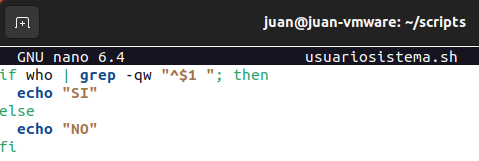
1. **Realizar un script llamado ‘usuarioconectado’ que retorna un SI si el primer parámetro coincide con algún usuario conectado o NO en caso contrario.**

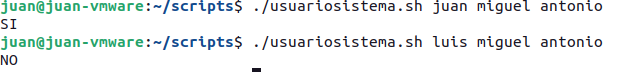
****

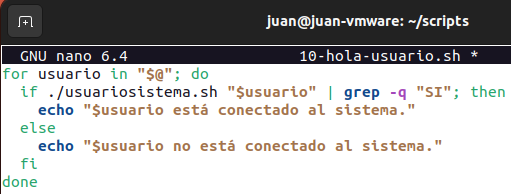
****

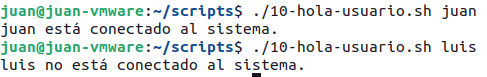
1. **Modificar el fichero ‘.bashrc’ para modificar el PATH y añadir la carpeta de estos ejercicios. Para ello añade la siguiente linea: export PATH=$PATH”:~/ruta\_carpeta\_ejercicios”**
2. **Modificar el script ‘05-hola-usuario.sh’ para que llame a ‘usuarioconectado’ (‘08-hola-usuario.sh’).**

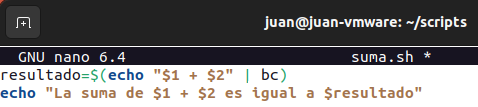
****

1. **Realizar un script llamado ‘usuariosistema’ que retorna un SI si el primer parámetro coincide con algún usuario del sistema o NO en caso contrario.**

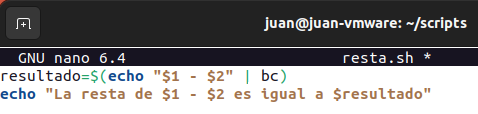
****

1. **Modificar el script ‘08-hola-usuario.sh’ para que llame a ‘usuariosistema’ (‘10-hola-usuario.sh’).**

****

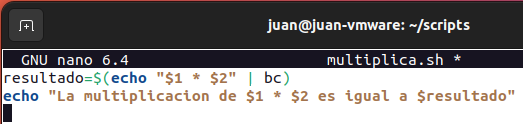
1. **Realizar un script llamado ‘suma’ que realice la suma de 2 parámetros introducidos (tendrá que poder sumar números decimales, como 2.2 + 3).**

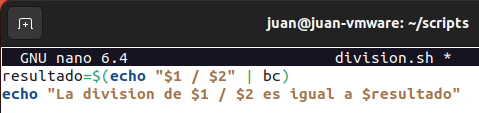
****

1. **Realizar un script llamado ‘resta’ que realice la resta de 2 parámetros introducidos (tendrá que poder restar números decimales, como 2.2 - 3).**

****

1. **Realizar un script llamado ‘multiplica’ que multiplique los 2 parámetros introducidos (tendrá que poder multiplicar números decimales, como 2.2 \* 3).**

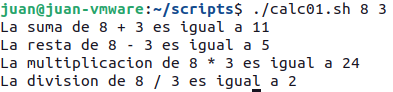
****

1. **Realizar un script llamado ‘division’ que realice la división de 2 parámetros introducidos (tendrá que poder sumar números decimales, como 2.2 / 3).**

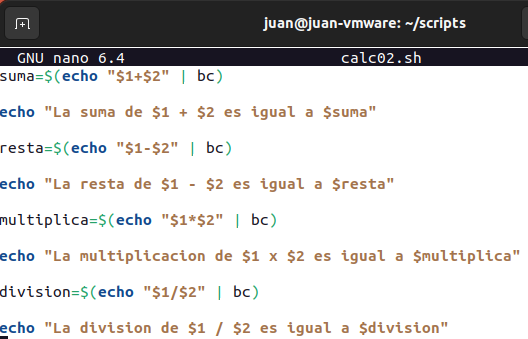
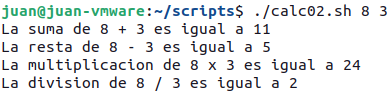
****

1. **Realizar un script llamado ‘calc01.sh’ que realice operaciones básicas entre 2 números llamando a cada uno de los scripts anteriormente creados (suma, resta, multiplicación y división).**

****

****

1. **Ídem pero sin llamar a los scripts (‘calc02.sh’).**

****

1. **Realizar un script llamado ‘citas-menu.sh’ que muestre un menú con las siguientes opciones:**

**1.Añadir cita nueva.**

**2.Buscar por nombre del paciente.**

**3.Buscar citas por hora inicial.**

**4.Buscar citas por hora final.**

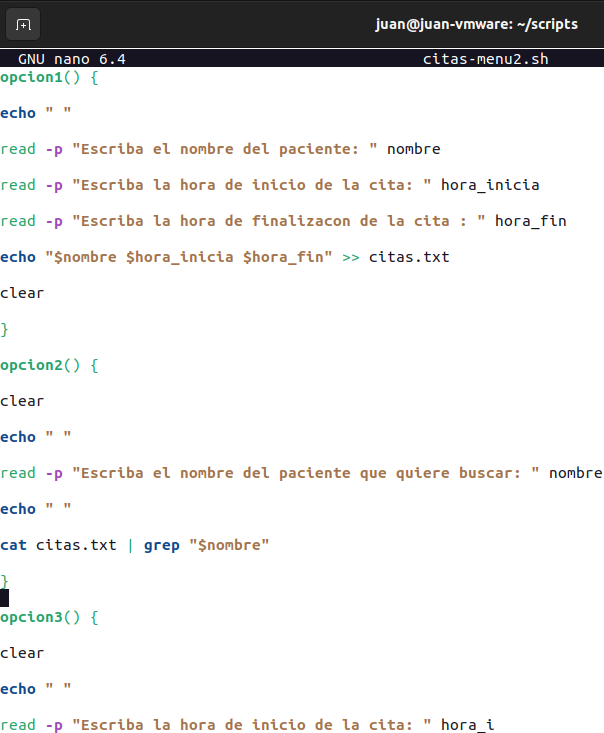
**5.Listar las citas ordenadas por nombre del paciente.**

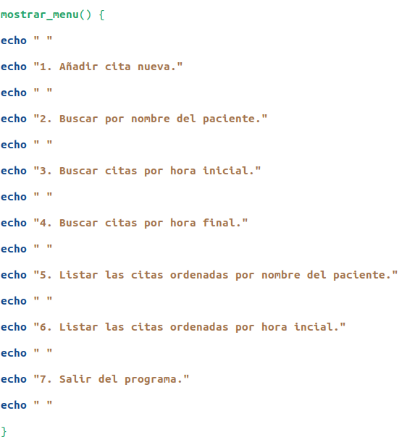
**6.Listar las citas ordenadas por hora inicial.**

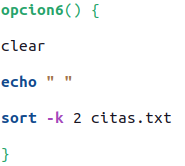
**7.Salir del programa.**

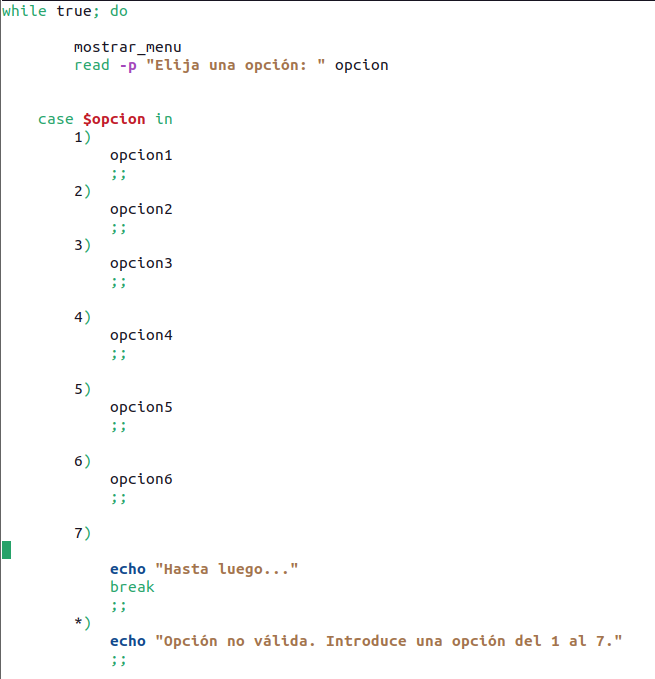
**Después, modifica el script citas.menu.sh al que llamaras citas.sh para que defina las funciones para realizar las opciones del menú.**

**NOTA: necesitaras un fichero citas.txt con el formato: nombrePaciente:horaInicio:horaFin**

****

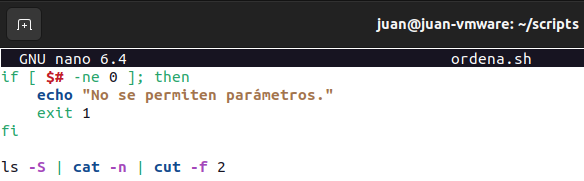
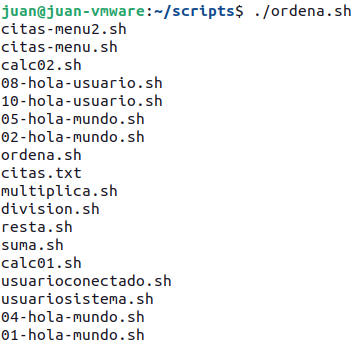
****

****

****

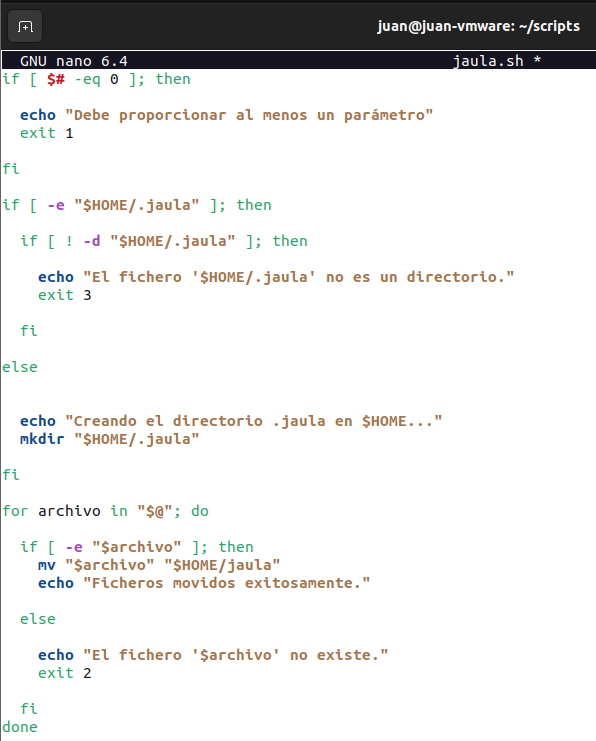
****

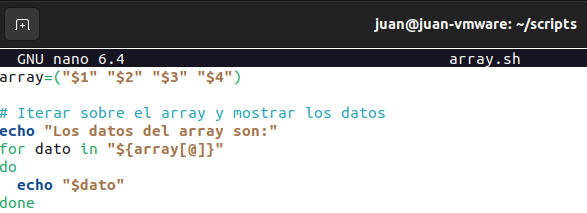
1. **Realizar un script llamado ‘ordena’ que liste el contenido del directorio actual ordenado por tamaño del archivo de menor a mayor. El listado sólo mostrará el nombre de los archivos y el número de línea correspondiente. En el caso de que se introduzca algún parámetro se mostrará el siguiente mensaje de error: “No se permiten parámetros.” y retornará un código de retorno igual a 1.**

****

****

1. **Realizar un script llamado ‘jaula’ que cree, sólo si no existe, el directorio jaula en la $HOME del usuario y mueva los ficheros pasados por parámetro a dicho directorio.** 
   1. **En el caso de que no se le pase ningún parámetro se mostrará el siguiente mensaje de error: “Hay que introducir al menos un parámetro.” y retornará un código de retorno igual a 1.**
   2. **En el caso de que algún fichero introducido por parámetro no exista se mostrará el siguiente mensaje de error: “El fichero ‘$FICHERO’ no existe.” y retornará un código de retorno igual a 2.**
   3. **Si el fichero . jaula existe en la $HOME del usuario pero no es un directorio mostrará el siguiente mensaje de error: “El fichero ‘$HOME/.jaula’ no es un directorio.” y retornará un código de retorno igual a 3.**

****

1. **Crear un script llamado ‘array.sh’ que declare un array, lo rellene con datos y luego itere sobre el mismo para mostrar los datos.**

